Pattern Recognition Letters 161 (2022) 67–73

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Rethinking class orders and transferability in class incremental learning

Chen He[a,b], Ruiping Wang[a,b,*], Xilin Chen[a,b]

[a] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[b] University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 18 June 2022
Revised 15 July 2022
Accepted 16 July 2022
Available online 19 July 2022

Edited by Jiwen Lu

Keywords:
Transferability
Class incremental learning
Class order

ABSTRACT

Class Incremental Learning (CIL), an indispensable ability for open-world applications such as service robots, has received increasing attention in recent years. Although many CIL methods sprouted out, researchers usually adopt default class orders, leaving the characteristics of different class orders less visited. In this paper, we rethink class orders in CIL from the following aspects: first, we show from preliminary studies that class orders do have an impact on the performance, and mainstream episodic memory-based CIL methods generally favor an interleaved way of arranging class orders; then, we interpret the phenomena above with transferability and propose transferability measures of class orders, which are in line with the method performance under different class orders; based on that, we propose a Class Order Search Algorithm (COSA) to obtain an optimal class order by finding which one has almost the highest transferability. Experiments on Group ImageNet and iNaturalist verify the importance of class orders in CIL methods, and demonstrate the effectiveness of our proposed transferability measures and COSA. These findings may help raise more attention to the hardly visited class orders in CIL.

1. Introduction

Learning is inherently an incremental process, and one may learn something earlier or later than another. According to psychology, the learning sequence of materials does have an impact on the learning performance, and it is usually believed that learning different materials alternatively (i.e. *interleaved learning*) is better than concentrating on one material until it is mastered and moving to the next one (i.e. *blocked learning*) [1,2]. To exemplify it, let us consider two sequences AABB and ABAB where A and B denote two different learning materials. Then, the conclusion above indicates that people who learn in an ABAB way perform better in these two tasks. The reasons are two-fold: on the one hand, being exposed to the same material constantly may get the learner customed to it and there is decreasing attention and knowledge gain; on the other, ABAB enables the learner to review the materials occasionally to alleviate forgetting. Probably because *interleaved learning* is more effective, the curriculum in schools or universities is arranged in an interleaved fashion.

Incremental Learning (IL), one of the most prosperous fields in machine learning that mimics the ongoing learning ability of humans, also faces a problem with the order of learning. To analyze it in a more pure setting, we focus on the most thriving and challenging subfield of IL called *Class-IL (CIL)*, which assumes that samples of one class or a bunch of classes arrive at a time. In CIL, the order of how classes arrive (i.e. class order) is seldom visited and researchers usually use the random or certain predefined class orders by default. Motivated by the fact in human learning that orders do have an impact on the learning performance, a natural question is whether a similar phenomenon also exists in CIL? Is an interleaved way of arranging class orders generally better (similar to human learning)?

To answer these questions, we first simulate the above-mentioned *interleaved learning* and *blocked learning* in CIL settings by leveraging two corresponding class orders denoted as *even* and *group* respectively (Fig. 1), where *even* means that the incoming classes at each incremental phase are evenly distributed over all superclasses, while *group* implies that the incoming classes at each incremental phase may come from the same superclass. Through preliminary studies, we almost constantly observe the superior performance of *even* for mainstream episodic memory-based CIL

* Corresponding author at: Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology CAS, Beijing 100190, China.
E-mail addresses: chen.he@vipl.ict.ac.cn (C. He), wangruiping@ict.ac.cn (R. Wang), xlchen@ict.ac.cn (X. Chen).

https://doi.org/10.1016/j.patrec.2022.07.014
0167-8655/

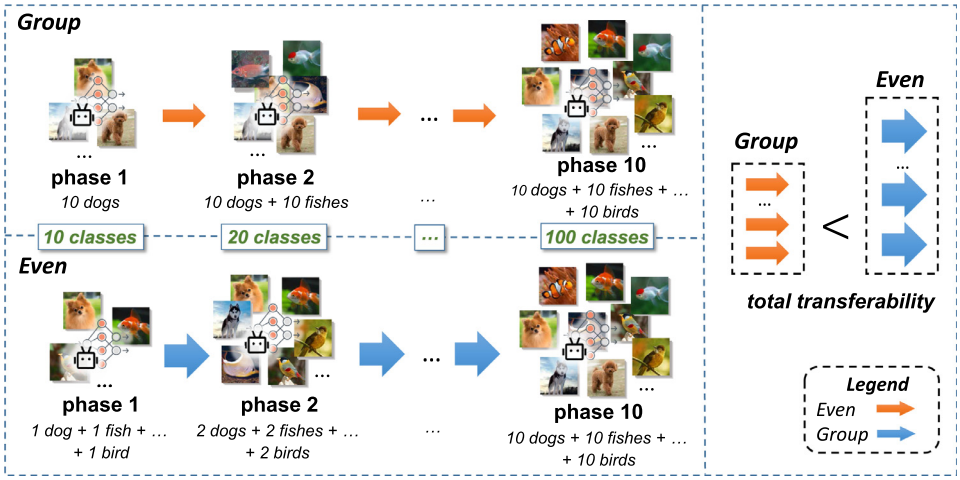

Fig. 1. Illustration of two typical class orders *group* and *even* (left). Both have the same target class group in the final incremental phase. For each incremental phase, there may be higher transferability (thicker arrow in the figure) between the previous and next batch of classes in *even* due to the higher similarity between these two batches. Thus, the total transferability in *even* is higher than *group* (right).

methods (Section 4.2), which is consistent with the superiority of *interleaving learning* in psychology mentioned above.

To gain a deeper understanding of such phenomena, we resort to tools of transferability. Our intuition is that seen classes at adjacent incremental phases for *even* are more similar and it is easier to transfer from one group to another, which probably accounts for the superiority of *even*. To verify it, we propose transferability measures defined on class orders to reflect the difficulty of continually transferring from old to new classes along the class order (Section 3.3), and the estimated transferability is in line with the above-mentioned performance of *even* and *group* (i.e. *even* with higher transferability outperforms *group*) (Section 4.3). Based on the transferability measures, the optimal class order[1] with almost the highest transferability can be obtained via a Class Order Search Algorithm (COSA) (Section 3.4), and the performance of episodic memory-based CIL methods under this searched class order can be on par or higher than those under *even*, which is a handcrafted class order that leads to the best performance observed so far on these datasets (Section 4.4). Further analyses by ablating the common techniques in these episodic memory-based CIL methods offer the reason why they favor *even* (Section 4.5). With all these findings, we discuss possible improvements and applications of the techniques introduced in this paper (Section 4.6), and call for more attention to the less visited problems of class orders in CIL.

2. Related works

Incremental learning Incremental learning (IL) [3,4], the ability of learning algorithms to continually incorporate new information without forgetting old knowledge, has received tremendous attention in the last few years [5,6]. In the large spectrum of IL, Task-IL, Domain-IL, and Class-IL (CIL) are what most researchers focus on, and CIL is generally believed to be the most difficult and realistic one of the three [7,8]. Thus, our work also revolves around CIL. The major problem that often co-occurs with CIL is catastrophic forgetting [9], where learning new information may completely disrupt old knowledge. While a plethora of IL works with novel mechanisms sprouted up in the last few years, what proved effective in CIL is still leveraging an additional memory to alleviate forgetting: either via generative memory [10–13] or episodic memory [14–19]. Our work focuses on the mainstream episodic memory-based CIL methods that store old exemplars since these methods are simple and effective. We analyze their characteristics under different class orders.

Orders in machine learning Existing works in machine learning analyze three types of orders: *sample orders*, *task orders*, and *class orders*. As for *sample orders*, curriculum learning [20] learns "easy" samples first for better convergence based on the expert knowledge of "easiness", and Self-Paced Learning [21] improves it by letting the model automatically learn "easiness" of the samples without any expert knowledge. As for *task orders*, active task selection [22] and task curriculum learning [23] are typical works that focus on arranging the task orders for better overall performance on all tasks either based on task relatedness or information maximization. This work [24] introduces the problem of task-order sensitivity and proposes an order-robust approach that decomposes the network parameters into shared and sparse task-adaptive parameters. As for *class orders*, there are few works [5,25] and the work of Masana et al. [25] is most related to ours that states that class orders may affect the performance of CIL methods. However, there are huge differences in the aim and implementation: that work [25] observes unsteady performance of CIL methods under many different class orders, and stresses the importance of using multiple class orders to test the CIL method's robustness. Our work mainly focuses on two typical class orders *even* and *group* that conform to *interleaved* and *blocked learning* in human learning, and observes almost consistent superiority of an interleaved way of arranging class orders. It inspires us to imitate the interleaving characteristic in *even* and search for the class order with almost the highest transferability, which is hardly covered by Masana et al. [25]. Moreover, we perform more in-depth analyses of why these methods favor *even* by ablating common techniques in these methods (Section 4.5).

Transferability Transferability, the difficulty of a model to transfer from one task to another, is fundamental to transfer learning [26] and other downstream tasks that rely on transfer learning including few-shot learning [27], incremental learning [28] etc. As long as the transferability among tasks is correctly estimated, we can know which tasks can be easily fine-tuned from a pre-trained model, or which source model is optimal to transfer onto a target task. Since transferability is usually defined between tasks, CIL seems a little irrelevant since there is no acknowledged concept of tasks in CIL. In this paper, we treat the classification of all seen classes at each incremental phase as a task. Therefore, a class order

[1] The optimal class order throughout this paper means the class order with the highest transferability instead of the class order that leads to optimal performance for CIL methods.

will lead to a sequence of correlated tasks where the classes may be overlapped among tasks. To estimate the transferability among tasks in CIL, we propose novel transferability measures to estimate the difficulty to transfer from the previous class group to the next one. The main reason for not using existing transferability measures [29,30] is that our proposed one is made of class-class distances, making it easier to perform a class order search algorithm to be elaborated in Section 3.

3. Method

As mentioned in Section 1, the experimental results that *even* outperforms *group* will be elaborated in Section 4.2 and we simply presume that the phenomena are known in this section. To interpret these phenomena, we resort to transferability measures defined on class orders. Specifically, we first define the transferability measure between two classes based on the visual or semantic distance,[2] then define the transferability measure between two consecutive tasks (i.e. class groups) using a "sigma-min" that aggregates the transferability among all class pairs, and finally sum them up over all incremental phases. With the transferability measure on class orders defined, we can obtain the optimal class order with the highest transferability by applying a search algorithm. We will start with the CIL formulation first.

3.1. CIL formulation

CIL assumes that samples of a class or a batch of classes arrive at a time. For simplicity, we assume that exactly K classes are added at a time and there are totally T class increments, which implies that the dataset has TK classes [14]. $X_{tr}^{(c)}$ and $X_{ts}^{(c)}$ are the training and test samples of class c ($c \in \{1,\dots,TK\}$) respectively. At time t ($t \in \{1,\dots,T\}$), the model needs to learn new classes $\{X_{tr}^{((t-1)K+1)},\dots,X_{tr}^{(tK)}\}$, and the objective is to achieve ideal classification results on the test set of the seen tK classes, i.e. $\{X_{ts}^{(1)},X_{ts}^{(2)},\dots,X_{ts}^{(tK)}\}$. For simplicity, we denote the label space at time t as $\mathcal{Y}_t$ which consists of the labels of all seen classes. Hence, we have $\mathcal{Y}_t \subset \mathcal{Y}_{t+1}$ ($t \in \{1,\dots,T-1\}$). Since discarding all old samples when learning new classes leads to severe catastrophic forgetting, researchers usually maintain an extra memory with a fixed budget M to store old exemplars (i.e. episodic memory-based) [14,15].

3.2. Class pair transferability measure

We use distance $d(i,j)$ to reflect the transferability between class i and j, and provide different choices as follows. Since semantically similar classes may share more commonalities and are easier to transfer to each other (e.g. two kinds of dogs), we leverage the Wu–Palmer distance [31] based on the WordNet hierarchy [32] that reflects the semantic relatedness between words:

$$d_{wup}(i,j) = 1 - \frac{2 \times depth(lcs(\boldsymbol{s}_i,\boldsymbol{s}_j))}{depth(\boldsymbol{s}_i)+depth(\boldsymbol{s}_j)} \tag{1}$$

In Eq. (1), s_i and s_j are the synsets[3] for class i and j respectively. $depth(\cdot)$ is the depth of the synset in the WordNet hierarchy, and $lcs(\cdot,\cdot)$ is the Lowest Common Ancestor (LCS) of the two given synsets. The Wu–Palmer distance takes both path distance in a taxonomy and class granularity into account, making it a reasonable hierarchical distance. Other semantic distances based on word embeddings or knowledge graphs can also be used, which is beyond the scope of this work.

Since the semantic distance does not consider image samples, it may be less accurate than visual distances that reflect the actual distribution. As for visual distances, we first choose a feature space that can be obtained via self-supervised learning or supervised learning (more information in Section 4), and then take the common assumption that the features of each class obey a multivariate Gaussian distribution. Thus, a broad spectrum of statistical distances between the feature distributions of two classes has easy-to-compute closed forms. For example, given the feature distributions of two classes $\mathcal{N}(\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$ and $\mathcal{N}(\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$, the 2-Wasserstein distance (WD) is defined as [33]:

$$d_{WD}(i,j) = \|\boldsymbol{\mu}_i-\boldsymbol{\mu}_j\|_2^2 + \|\boldsymbol{\Sigma}_i^{1/2}-\boldsymbol{\Sigma}_j^{1/2}\|_F^2 \tag{2}$$

Wasserstein distance is commonly employed to solve the optimal transport problem [34,35] and has been adopted to estimate the transferability between two datasets [36] etc. Moreover, inspired by the success of a combination of Mahalanobis Distance (MD) between class means and Log-Euclidean Distance (LED) between covariance matrices in image set classification [37], we define MD-LED distance by adding them[4]:

$$\begin{aligned} d_{MD-LED}(i,j) = {} & (\boldsymbol{\mu}_i-\boldsymbol{\mu}_j)^T(\boldsymbol{\Sigma}_i^{-1}+\boldsymbol{\Sigma}_j^{-1})(\boldsymbol{\mu}_i-\boldsymbol{\mu}_j) \\ & + \|\log(\boldsymbol{\Sigma}_i)-\log(\boldsymbol{\Sigma}_j)\|_F^2 \end{aligned} \tag{3}$$

In Eq. (3), $\log(\boldsymbol{\Sigma})$ is defined as $\log(\boldsymbol{\Sigma}) = \boldsymbol{U}\log(\boldsymbol{\Lambda})\boldsymbol{U}^T$, where $\boldsymbol{\Lambda}$ and $\boldsymbol{U}$ are the diagonal matrix of the eigenvalue logarithms and the orthogonal matrix in eigen-decomposition respectively.

The reasons for choosing the two visual distances above are: WD has been adopted in transfer learning [36] and LED has been used in domain adaptation [38], which are both related to transferability. Other distances related to transfer learning or domain adaption (e.g. MMD [39]) can also be used.

3.3. Class order transferability measure

After defining the transferability measure between two classes, the transferability measure between class groups $\mathcal{Y}_t$ and $\mathcal{Y}_{t+1}$ denoted as $\mathcal{S}_{t\to t+1}$ can be defined as the following "sigma-min" form similar to the Chamfer distance:

$$\mathcal{S}_{t\to t+1} = \sum_{j\in\mathcal{Y}_{t+1}} \min_{i\in\mathcal{Y}_t} d(i,j) \tag{4}$$

In Eq. (4), i and j stands for a class from the tth class group $\mathcal{Y}_t$ and the $(t+1)$th class group $\mathcal{Y}_{t+1}$ respectively. $d(i,j)$ can be chosen from the distances defined in Section 3.2 (i.e. Eqs. (1)–(3)). The intuition is that for each new class we find the most similar old class and calculate the distance between them, then we sum the distances over all new classes. Such a form is easy to compute and the corresponding algorithm to solve the optimal class order to be mentioned next is rather simple.

Consequently, the class order transferability measure $\mathcal{S}$ is:

$$\mathcal{S} = \textstyle\sum_{t=1}^{T-1} \mathcal{S}_{t\to t+1} \tag{5}$$

Eq. (5) simply sums up Eq. (4) over $T-1$ time steps. Since $\mathcal{S}$ is a distance-based measure, a smaller value of S indicates higher transferability of the class order.

3.4. Class order search algorithm (COSA)

Given the class order transferability measure in Eq. (5), the problem to find the optimal class order can be formulated as:

[2] Strictly speaking, the distance reflects the non-transferability.

[3] A group of synonymous words.

[4] Here, we simply add MD and LED. Since MD and LED can have different magnitudes, it would be better to assign different weights when adding them.

$$\begin{aligned} \underset{\mathcal{Y}_1,\ldots,\mathcal{Y}_{T-1}}{\arg\min} \quad & \sum_{t=1}^{T-1}\sum_{j\in\mathcal{Y}_{t+1}}\min_{i\in\mathcal{Y}_t} d(i,j) \\ \text{s.t.} \quad & \mathcal{Y}_t \subset \mathcal{Y}_{t+1} \quad (1\le t\le T-1) \\ & |\mathcal{Y}_t| = tK \quad (1\le t\le T) \end{aligned} \tag{6}$$

Note that $\mathcal{Y}_T$ is fixed, which is the group of all classes in the dataset. If we know the previous class groups $\mathcal{Y}_1,\ldots,\mathcal{Y}_{T-1}$, the class order can be uniquely determined.[5] Since there are multiple variables of class groups $\mathcal{Y}_t$ $(t\in 1,\ldots,T)$ that are mutually constrained and lead to numerous combinations, the global optimum is intractable. Inspired by the fact that the final class group $\mathcal{Y}_T$ is fixed, we can take a reverse greedy search which is to iteratively determine the optimal $\mathcal{Y}_t$ given $\mathcal{Y}_{t+1}$:

$$\begin{aligned} \underset{\mathcal{Y}_t}{\arg\min} \quad & \sum_{j\in\mathcal{Y}_{t+1}}\min_{i\in\mathcal{Y}_t} d(i,j) \\ \text{s.t.} \quad & \mathcal{Y}_t \subset \mathcal{Y}_{t+1} \quad (1\le t\le T-1) \\ & |\mathcal{Y}_t| = tK \quad (1\le t\le T) \end{aligned} \tag{7}$$

Interestingly, such a formulation is almost the same as the objective of k-medoid [40], which is a clustering method similar to k-means but chooses actual points as centers. The centroids solved by k-medoids are exactly the optimal or at least near-optimal $\mathcal{Y}_t$ for Eq. (7).

Note that when $T=2$, Eq. (6) degenerates into Eq. (7) and the reverse greedy search gives the global optimum. However, when $T>2$, the reverse greedy search only yields a local optimum of Eq. (6). For those interested in finding a solution closer to the global optimum when $T>2$, a beam search [41] can be employed. However, in our experiments, we find that the solution of the reverse greedy search is good enough, and we do not visit other algorithms in this work and leave it for future works.

4. Experiments

4.1. Experimental setup

Methods We compare mainstream episodic memory-based CIL methods iCaRL [14], End-to-End Incremental Learning (EEIL) [15], Large Scale Incremental Learning (LSIL) [16], IL2M [17], Weight Aligning (WA) [42], post-scaling [19]. The implementation details are in the supplementary material. The source code is available at https://github.com/TonyPod/RethinkingClassOrder and http://vipl.ict.ac.cn/zygx/dm/.

Datasets We use two datasets Group ImageNet and Group iNaturalist. Group ImageNet is a 100-class subset of ImageNet 1K [43] introduced by He et al. [19]. It covers 10 superclasses and each superclass has exactly 10 classes. Similar to Group ImageNet, Group iNaturalist is a 81-class subset of iNaturalist [44] that covers 9 superclasses and each superclass has exactly 9 classes. The superclasses in the two datasets are shown in Fig. 2. The image resolution is 64×64 for both datasets. More details of these two datasets are in the supplementary material.

Evaluation protocol As for the incremental protocol, we use 20×5 to imply that there are 5 class increments and each class increment adds 20 new classes. Unless otherwise specified, we use 10×10 for Group ImageNet and 9×9 for Group iNaturalist. Note that *even* or *group* is a type of class order that can have different actual class orders by random shuffling. We report the accuracies of different methods in the final incremental phase for different class orders. To reduce the randomness of a single run, the reported results are averaged over 5 different actual orders of the corresponding type. For our searched class order, since it is uniquely determined, we do not average the results over 5 different orders as *even* or *group* does.

[5] The order of classes inside a class group is unimportant.

4.2. Class orders do matter in CIL

The final classification accuracies of recent CIL methods under two class orders *even* and *group* on Group ImageNet are shown in Table 1. From the results, it can be seen that *even* almost constantly outperforms *group* for different incremental protocols. Also, if there are more class increments, the *superiority of even* is more obvious ("10×10" vs. "50×2"). In the supplementary material, we display more results on Group iNaturalist and the results via different hyper-parameters on Group ImageNetwhere we still have the conclusion that *even* almost consistently outperforms *group*. These results are reminiscent of the phenomenon from psychology mentioned in Section 1 that *interleaved learning* (i.e. ABAB) is better than *blocked learning* (i.e. AABB), where the class order of *even* is "dog-1, fish-1, ..., bird-1, dog-2, fish-2, ..., bird-2, ... " similar to the pattern "ABAB" and *group* is "dog-1, ..., dog-10, fish-1, ..., fish-10, ... " similar to the pattern "AABB". In Section 4.5, we provide further analyses of why these methods favor *even*.

4.3. Relationship with transferability

In Section 3.3, we have defined transferability measures for class orders. Here, we show the estimated transferability of *even* and *group* to see if they are in line with the performance of CIL methods mentioned in Section 4.2. As for the semantic similarity, it can be easily calculated by using the class labels. As for the visual similarities, we use two different types of proxy networks to extract features: supervised (Sup.) and self-supervised (SS). The supervised network is trained on images of all classes and corresponding labels in the dataset, whereas the self-supervised network is only trained on images of all classes via the simple proxy task of rotation prediction [45]. The estimated transferability on Group ImageNet is summarized in Table 2. Note that the estimated transferability based on different distances are non-comparable—Only the estimated transferability of different class orders using the same feature space and distance function is comparable (i.e. the values in the same row are comparable). It can be seen that *even* generally has higher transferability (i.e. lower distance) than *group* for almost all transferability measures and incremental protocols. Also, such a phenomenon is more obvious when there are more class increments ("10×10" vs. "50×2"), which is in line with the conclusion in Section 4.2 that the superiority of *even* is more obvious under these situations. In the supplementary material, we show the results on Group iNaturalist and still observe that *even* generally has higher transferability. Thus, we connect the performance under different class orders with the transferability measures, which lays the foundation for COSA.

4.4. Effectiveness of COSA

In Section 4.2, we show from experiments that *even* performs better than *group* and is among the optimal candidates, thus we hope that our searched class order can have comparable or higher transferability than the handcrafted *even*. In Table 2, we display the estimated transferability of the class order obtained by COSA (denoted as *greedy* since it takes a greedy search). It can be seen that for most cases *greedy* has higher transferability than *even*, which verifies the effectiveness of COSA. We recommend that the readers see the visualization of different orders that shows the effectiveness of COSA in the supplementary material. Also, the performance of CIL methods under *greedy* is comparable to or better than that under *even* (Table 3), which again verifies the effectiveness of COSA.

From the table, it can be seen that *greedy* does not always outperform *even*, and the reasons are two-fold: the reverse greedy

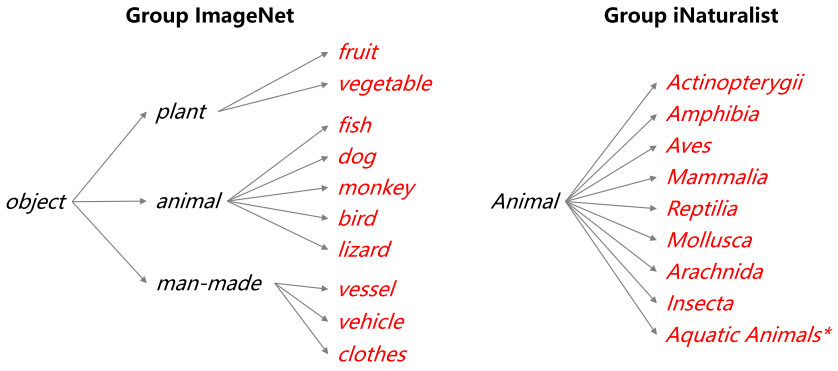

Fig. 2. Simplified hierarchies of Group ImageNet and Group iNaturalist. The classes in red (i.e. leaf nodes in the hierarchies) are the chosen superclasses, each of which has 10 and 9 subclasses for Group ImageNet and Group iNaturalist respectively. "Aquatic Animals*" means that most of the subclasses are aquatic animals. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Classification accuracies (%) of methods with the *even* and *group* class order on Group ImageNet. Each result is averaged by 5 different actual class orders of the corresponding class order type, and we denote the standard deviation. The class order that leads to the better accuracy for different methods under three incremental protocols is highlighted in **bold**.

Method	10 × 10		20 × 5		50 × 2	
	even	group	even	group	even	group
iCaRL [14]	**41.47 ± 0.46**	39.40 ± 1.52	**47.33 ± 0.57**	44.38 ± 0.47	**53.89 ± 0.40**	52.80 ± 1.21
EEIL [15]	**45.79 ± 0.35**	43.69 ± 0.46	**54.35 ± 0.21**	52.68 ± 0.28	**65.20 ± 0.71**	64.05 ± 0.76
LSIL [16]	**42.56 ± 1.35**	20.86 ± 4.41	**54.04 ± 0.54**	36.94 ± 3.14	**63.95 ± 0.58**	61.05 ± 1.53
IL2M [17]	**35.72 ± 0.67**	33.15 ± 0.82	**42.75 ± 1.30**	38.98 ± 0.60	**54.85 ± 0.77**	53.81 ± 0.79
WA [42]	**46.64 ± 0.94**	43.88 ± 1.06	**53.88 ± 0.64**	49.87 ± 1.60	**63.44 ± 0.55**	62.82 ± 0.65
Post-scaling [19]	**47.42 ± 0.44**	45.05 ± 0.85	**55.86 ± 0.44**	52.86 ± 0.90	**66.40 ± 0.51**	65.49 ± 0.76

Table 2
Estimated transferability of our searched class order *greedy, even* and *group* on Group ImageNet using different CIL protocols. "Sup./SS" is short for the supervised/self-supervised feature. The lower, the better.

Transferability measure based on	10 × 10			20 × 5			50 × 2		
	greedy	even	group	greedy	even	group	greedy	even	group
WD (Sup.)	5.24×10^3	5.25×10^3	6.18×10^3	4.41×10^3	4.55×10^3	5.34×10^3	2.69×10^3	2.73×10^3	3.13×10^3
WD (SS)	2.79×10^1	2.81×10^1	3.38×10^1	2.21×10^1	2.34×10^1	2.99×10^1	1.12×10^1	1.27×10^1	1.48×10^1
MD-LED (Sup.)	3.27×10^4	3.37×10^4	3.95×10^4	2.90×10^4	2.88×10^4	3.38×10^4	1.74×10^4	1.72×10^4	1.96×10^4
MD-LED (SS)	4.17×10^2	4.11×10^2	4.83×10^2	3.42×10^2	3.55×10^2	4.28×10^2	2.11×10^2	2.07×10^2	2.27×10^2
Wu–Palmer	1.00×10^1	1.00×10^1	1.32×10^1	8.45	8.73	1.11×10^1	4.90	5.31	6.34

Table 3
Classification accuracies (%) of methods under the searched class orders by COSA (i.e. *greedy*) based on different transferability measures on Group ImageNet. The protocol is 10 × 10. We average the accuracies under different *greedy* in the "average" column. We list the accuracies under *even* and *group* as references.

Method	Searched order based on						Reference	
	WD (Sup.)	WD (SS)	MD-LED (Sup.)	MD-LED (SS)	Wu–Palmer	average	even	group
iCaRL [14]	41.94	42.00	40.38	40.00	40.64	40.99	41.47	39.40
EEIL [15]	46.76	46.30	44.72	46.26	45.56	45.92	45.79	43.69
LSIL [16]	39.34	42.78	41.06	39.88	40.68	40.75	42.56	20.86
IL2M [17]	35.56	37.12	36.28	35.72	35.94	36.12	35.72	33.15
WA [42]	45.90	45.60	45.12	45.96	45.86	45.69	46.64	43.88
Post-scaling [19]	46.74	47.96	46.84	47.78	47.18	47.30	47.42	45.05

search in COSA only gives a local optimum when $T > 2$, making the class order a suboptimal one; the transferability may be not the only factor in determining the performance, which will be discussed more detailedly in Section 4.6. The results for a 20 × 5/50 × 2 incremental protocol on Group ImageNet and a 9 × 9/27 × 3 incremental protocol on Group iNaturalist are shown in the supplementary material, which leads to the same conclusion. Also, among these transferability measures, WD generally gives better classification performance. The reasons are two-fold: (1) As noted in Footnote 4, MD-LED is implemented by simply adding MD and LED without weighting factors, which may incur errors since these two terms have quite different magnitudes. The weighting factors should be carefully chosen, which is left for future works; (2) The Wu–Palmer distance does not consider image samples, which may be "less accurate than visual distances that reflect the actual distribution" as mentioned in Section 3.2.

4.5. Further analyses of why class order matters

The aforementioned analyses of transferability are simply based on the dataset and are model-independent. Thus, we may hypothesize that for a black-box CIL method, it is more likely to still fa-

Table 4
Classification accuracies (%) of ablating the techniques in CIL methods on Group ImageNet. Each model is trained for 70 epochs at each incremental phase. Each result is averaged over 5 different class orders of the corresponding order type. "w/ distill" means using the distillation loss. "w/ finetune" means that the new model is fine-tuned from the old one. "Base" is the variant without the two aforementioned techniques. The number inside "()" is the improvement of accuracy over "Base". IL2M does not have the distillation loss, thus there is a "/" in certain elements of the table.

Method	Even				Group			
	Base	w/ distill	w/ finetune	w/ both	Base	w/ distill	w/ finetune	w/ both
iCaRL [14]	30.92	36.60 (+5.68)	37.50 (+6.58)	41.47 (+10.55)	30.46	35.25(+4.79)	34.79 (+4.33)	39.40 (+8.94)
EEIL [15]	32.92	40.57 (+7.65)	37.48 (+4.56)	45.79 (+12.87)	32.31	38.64 (+6.33)	36.62 (+4.31)	43.69 (+11.38)
LSIL [16]	30.73	38.28 (+7.55)	36.40 (+5.67)	42.56 (+11.83)	22.41	16.10 (−6.31)	28.90 (+6.49)	20.86 (−1.55)
IL2M [17]	28.68	/	35.72 (+7.04)	/	26.56	/	33.15 (+6.59)	/
WA [42]	30.00	43.15 (+13.15)	32.44 (+2.44)	46.64 (+16.64)	30.96	40.34 (+9.38)	34.76 (+3.80)	43.88 (+12.92)
Post-scaling [19]	33.44	42.71 (+9.27)	39.14 (+5.70)	47.42 (+13.98)	32.31	38.88 (+6.57)	36.48 (+4.17)	45.05 (+12.74)

vor *even*. However, strictly speaking, the performance under different class orders still depends on the characteristics of the learner. Thus, we offer further analyses to find out what commonalities make these methods favor *even*.

Apart from IL2M that does not have the *distillation loss*, all other methods share two common techniques that are related to knowledge transfer: *distillation loss* that originates in Li and Hoiem [28] and *fine-tuning* from the old model instead of training the new model from scratch. Thus, we ablate these two buildings blocks from these methods to observe their behavior under *even* and *group*. The results are shown in Table 4. It can be seen that "Base" of almost all methods under *even* and *group* does not differ too much in performance by comparing with "w/ distill" or "w/ finetune" except LSIL, which indicates that the *distillation loss* and *fine-tuning* are more sensitive to class orders. By scrutinizing the improvements of "w/ distill" and "w/ finetune" over "Base", we find that the phenomenon that the *distillation loss* favors *even* is more obvious. The reason is that the *distillation loss* forces the responses of the samples on the new model to be similar to those on the old model, which is a kind of review to alleviate forgetting. Therefore, the choice of the samples is rather important: if the samples are diverse, all previous classes can be reviewed. Since in *even* the samples are spread over all superclasses, it provides a better review of old knowledge than *group* where samples are concentrated inside only a few superclasses. Consequently, the *distillation loss* may be the important cause of the large performance gap for these methods under different class orders. Thus, it is recommended that the form of the *distillation loss* can be adapted, or the weighting factor of the *distillation loss* can be dynamically adjusted based on the transferability between the previous and next batch of classes. It may lead to more steady behavior of these methods under different class orders.

4.6. Discussions

Superiority of even. Although we make connections between the superiority of *interleaving learning* (i.e. the interleaving effect [2]) in psychology (Section 1) and the superiority of *even* in CIL (Section 4.2), it should be noted that experiments of *spacing effect* are mainly verified via *recall* or *relearning*, instead of *recognition* as in CIL. Therefore, the conclusions from human learning and machine learning may not be 100% consistent due to a difference in the experimental setting. However, we still encourage further collaborations between these two fields, which would bring more insights.

Transferability and performance. Although we draw connections between transferability and performance in CIL, we should note that too much transferability may also lead to a performance drop. For example, a new class *Alaskan Malamute* and an old class *husky* are two visually similar dogs. Thus, there is higher transferability between these two classes, but they are more likely to be confused by the model. This phenomenon indicates that transferability is probably not the only factor to reflect the classification performance, which requires more studies. In the supplementary material, we further discuss the potential applications of the proposed transferability measures and COSA.

5. Conclusion

In this paper, we revisit class orders in Class Incremental Learning (CIL). Specifically, we show from experiments that mainstream episodic memory-based CIL methods favor *even* class orders, which is in line with the superiority of *interleaved learning* in psychology. Then, we draw connections between the performance of these methods with our proposed transferability measures defined on class orders, where higher transferability is correlated with better performance. The transferability measures can also be indicators used to search for the optimal class order by finding which one has the highest transferability. Future works are further improving the transferability measures and applying the class order search algorithm to real-world scenarios mentioned in the discussion section.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is partially supported by Natural Science Foundation of China under contracts nos. U21B2025, U19B2036, 61922080, and National Key R&D Program of China no. 2021ZD0111901.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2022.07.014

References

[1] D. Rohrer, R.F. Dedrick, S. Stershic, Interleaved practice improves mathematics learning, J. Educ. Psychol. 107 (3) (2015) 900.
[2] S.C. Pan, The interleaving effect: mixing it up boosts learning, Sci. Am. 313 (2) (2015).
[3] R.J. Solomonoff, A system for incremental learning based on algorithmic probability, in: Proceedings of the Sixth Israeli Conference on Artificial Intelligence, Computer Vision and Pattern Recognition, 1989, pp. 515–527.
[4] A. Gepperth, B. Hammer, Incremental learning algorithms and applications, European Symposium on Artificial Neural Networks (ESANN), 2016.
[5] M. Masana, X. Liu, B. Twardowski, M. Menta, A.D. Bagdanov, J. van de Weijer, Class-incremental learning: survey and performance evaluation, arXiv preprint arXiv:2010.15277(2020).
[6] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: defying forgetting in classification tasks, IEEE TPAMI 44 (7) (2021) 3366–3385.
[7] Y.-C. Hsu, Y.-C. Liu, A. Ramasamy, Z. Kira, Re-evaluating continual learning scenarios: acategorization and case for strong baselines, arXiv preprint arXiv:1810.12488(2018).

[8] G.M. van de Ven, A.S. Tolias, Three scenarios for continual learning, arXiv preprint arXiv:1904.07734(2019).
[9] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of Learning and Motivation, Elsevier, 1989, pp. 109–165.
[10] H. Shin, J.K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, in: NeurIPS, 2017, pp. 2994–3003.
[11] C. He, R. Wang, S. Shan, X. Chen, Exemplar-supported generative reproduction for class incremental learning, in: British Machine Vision Conference (BMVC), 2018, p. 98.
[12] C. Wu, L. Herranz, X. Liu, J. van de Weijer, B. Raducanu, et al., Memory replay GANs: learning to generate new categories without forgetting, in: NeurIPS, 2018, pp. 5962–5972.
[13] O. Ostapenko, M. Puscas, T. Klein, P. Jahnichen, M. Nabi, Learning to remember: a synaptic plasticity driven framework for continual learning, in: CVPR, 2019, pp. 11321–11329.
[14] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, iCaRL: incremental classifier and representation learning, in: CVPR, 2017, pp. 2001–2010.
[15] F.M. Castro, M.J. Marín-Jiménez, N. Guil, C. Schmid, K. Alahari, End-to-end incremental learning, in: ECCV, 2018, pp. 233–248.
[16] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Y. Fu, Large scale incremental learning, in: CVPR, 2019, pp. 374–382.
[17] E. Belouadah, A. Popescu, IL2M: class incremental learning with dual memory, in: ICCV, 2019, pp. 583–592.
[18] Y. Liu, Y. Su, A.-A. Liu, B. Schiele, Q. Sun, Mnemonics training: multi-class incremental learning without forgetting, in: CVPR, 2020, pp. 12245–12254.
[19] C. He, R. Wang, X. Chen, A tale of two CILs: the connections between class incremental learning and class imbalanced learning, and beyond, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop on Continual Learning, 2021, pp. 233–248.
[20] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: International Conference on Machine Learning (ICML), 2009, pp. 41–48.
[21] M.P. Kumar, B. Packer, D. Koller, Self-paced learning for latent variable models, in: NeurIPS, 2010, pp. 1189–1197.
[22] P. Ruvolo, E. Eaton, Active task selection for lifelong machine learning, in: AAAI Conference on Artificial Intelligence (AAAI), 2013, pp. 862–868.
[23] A. Pentina, V. Sharmanska, C.H. Lampert, Curriculum learning of multiple tasks, in: CVPR, 2015, pp. 5492–5500.
[24] J. Yoon, S. Kim, E. Yang, S.J. Hwang, Scalable and order-robust continual learning with additive parameter decomposition, ICLR, 2020.
[25] M. Masana, B. Twardowski, J. Van de Weijer, On class orderings for incremental learning, in: International Conference on Machine Learning (ICML) Workshop on Continual Learning, 2020.
[26] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2009) 1345–1359.
[27] Q. Sun, Y. Liu, T.-S. Chua, B. Schiele, Meta-transfer learning for few-shot learning, in: CVPR, 2019, pp. 403–412.
[28] Z. Li, D. Hoiem, Learning without forgetting, in: ECCV, 2016, pp. 614–629.
[29] A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C.C. Fowlkes, S. Soatto, P. Perona, Task2Vec: task embedding for meta-learning, in: ICCV, 2019, pp. 6430–6439.
[30] A.T. Tran, C.V. Nguyen, T. Hassner, Transferability and hardness of supervised classification tasks, in: ICCV, 2019, pp. 1395–1405.
[31] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Annual Meeting of the Association for Computational Linguistics (ACL), 1994, pp. 133–138.
[32] G.A. Miller, WordNet: a lexical database for english, Commun. ACM 38 (11) (1995) 39–41.
[33] D.C. Dowson, B.V. Landau, The Fréchet distance between multivariate normal distributions, J. Multivar. Anal. 12 (3) (1982) 450–455.
[34] Y. Rubner, C. Tomasi, L.J. Guibas, A metric for distributions with applications to image databases, in: ICCV, IEEE, 1998, pp. 59–66.
[35] Y. Rubner, C. Tomasi, L.J. Guibas, The earth mover's distance as a metric for image retrieval, IJCV 40 (2) (2000) 99–121.
[36] D. Alvarez-Melis, N. Fusi, Geometric dataset distances via optimal transport, arXiv preprint arXiv:2002.02923(2020).
[37] W. Wang, R. Wang, Z. Huang, S. Shan, X. Chen, Discriminant analysis on Riemannian manifold of gaussian distributions for face recognition with image sets, in: CVPR, 2015, pp. 2048–2057.
[38] Y. Wang, W. Li, D. Dai, L. Van Gool, Deep domain adaptation by geodesic distance minimization, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 2651–2657.
[39] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.-P. Kriegel, B. Schölkopf, A.J. Smola, Integrating structured biological data by kernel maximum mean discrepancy, Bioinformatics 22 (14) (2006) 49–57.
[40] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., 1988.
[41] M. Freitag, Y. Al-Onaizan, Beam search strategies for neural machine translation, in: Annual Meeting of the Association for Computational Linguistics (ACL), 2017, pp. 56–60.
[42] B. Zhao, X. Xiao, G. Gan, B. Zhang, S.-T. Xia, Maintaining discrimination and fairness in class incremental learning, in: CVPR, 2020, pp. 13208–13217.
[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image database, in: CVPR, 2009, pp. 248–255.
[44] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S. Belongie, The iNaturalist species classification and detection dataset, in: CVPR, 2018, pp. 8769–8778.
[45] S. Gidaris, P. Singh, N. Komodakis, Unsupervised representation learning by predicting image rotations, ICLR, 2018.